INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

DHT-OLSR

Emmanuel Baccelli – Thomas Zahn – Jochen Schiller

N° 6194

Mai 2007

Thème COM

Rapport de recherche

ISSN 0249-6399

DHT-OLSR

E. Baccelli [1], T. Zahn [2], J. Schiller[3]

Thème COM – Systèmes communicants
Projets Hipercom, Arles

Rapport de recherche n° 6194 – Mai 2007 - 11 pages

Abstract: Self-organized networking is foreseen as an important component in the Internet's near-future architecture. An essential challenge concerning the integration of this new component is the accomplishment of scalable and efficient ad hoc routing. This report proposes a new solution in this space, DHT-OLSR, based on OLSR enhanced with dynamic clustering and distributed hash table (DHT) routing. We believe that such a protocol can provide an architecture that may introduce a gradual transition from traditional IP routing towards scalable IP MANET routing.

Keywords: network, mobile, ad hoc, routing, IP, DHT, OLSR, MADPastry

[1] INRIA FUTURS, France – emmanuel.baccelli@inria.fr
[2] INRIA Rocquencourt, France – thomas.zahn@inria.fr
[3] Frei Universität Berlin, Germany – schiller@inf.fu-berlin.de

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex (France)
Téléphone : +33 1 39 63 55 11 — Télécopie : +33 1 39 63 53 30

DHT-OLSR

Résumé: L'auto-organisation est considérée comme un élément important de l'architecture Internet dans un futur proche. Un défi majeur concernant l'intégration de cet élément est l'accomplissement du routage mobile ad hoc à grande échelle. Ce rapport propose une nouvelle solution dans ce domaine, DHT-OLSR.

Mots clés: réseau, ad hoc, mobile, routage, IP, DHT, OLSR, MADPastry

1 Introduction

Self-organized networking is foreseen as an important component in the Internet's near-future architecture. An essential challenge concerning the integration of this new component is the accomplishment of scalable and efficient mobile ad hoc IP routing.

While current MANET routing protocols perform well in small topologies, they generally fail to provide a functional ad hoc network when the number of nodes involved in the MANET increases. Proactive approaches (such as OLSR [4]), as well as reactive approaches (such as AODV [8] or Dymo [3]), require the use of network-wide control signaling that quickly saturates the available bandwidth when nodes participating in the MANET are too numerous and/or too mobile.

In order to overcome this problem, specific mechanisms have to be used in conjunction with usual MANET routing. However, additional requirements are to be considered for compatibility with the Internet architecture. In particular, IP routers are basically supposed to either (i) directly forward packets as they are received, or else (ii) drop them if no appropriate route is currently available. This philosophy disqualifies some mechanisms a priori (including reactive routing), and must be taken into account in the design of a solution for scalable mobile ad hoc IP networking.

One interesting approach is the cross-fertilization between the fields of mobile ad hoc routing and peer-to-peer networking since both fields are indeed dealing with similar issues, such as decentralized network formation and maintenance. For instance, specific DHT key-based approaches [2][5][7][12] can provide very efficient routing in large scale MANETs. However, we believe that it is unrealistic to assume that such key-based routing will directly replace traditional IP routing. This paper, thus, introduces and evaluates a hybrid solution, namely DHT-OLSR, based on dynamic OLSR clustering enhanced with distributed hash tables (DHT). This solution uses both the properties of efficient key-based unicast routing to scale as desired, and the properties of OLSR routing to naturally integrate with the Internet infrastructure. Therefore, this protocol provides an architecture that may introduce a gradual transition from traditional IP routing towards scalable IP MANET routing.

The remainder of the paper is organized as follows. The next two sections will briefly overview the principles of OLSR routing on one hand, and of MADPastry's key-based routing. The following sections will then describe in detail the functioning of DHT-OLSR, followed by a section providing experimental results. Before concluding this paper, we will briefly revisit related work.

2 The OLSR Protocol

In this section, we briefly outline OLSR, giving the essential facts about the protocol that are interesting in the context of this paper. For further details on OLSR, or on its performance characteristics, see [4].
As a proactive link-state routing protocol, OLSR (Optimized Link State Routing) employs the periodic exchange of control messages in order to accomplish topology discovery and maintenance. This exchange results in a topology map being present in each node in the network, from which a routing table can be constructed.
Basically, OLSR employs two types of control messages: HELLO messages and TC (Topology Control) messages. HELLO messages have local scope and are exchanged periodically between neighbor nodes only, essentially tracking the status of links between neighbors. On the other hand, TC messages have larger scope and are emitted periodically to diffuse link-state information throughout the entire network. This operation of diffusing a message to the entire network – also called flooding – is optimized in OLSR with a mechanism called MPR flooding (MPR stands for Multi-Point Relay, see [9] for more details on this OLSR-specific technique).
This optimization reduces drastically the cost of performing a flooding operation, through having each node select a minimal set of "relay nodes" (called MPRs), responsible for relaying flooded packets. As shown in Fig. 1, from the local point of view of a node flooding a packet – i.e. the center node in the figure – this corresponds to only the minimal number of neighbors (the black nodes) relaying the broadcast, instead of basically all the neighbors.

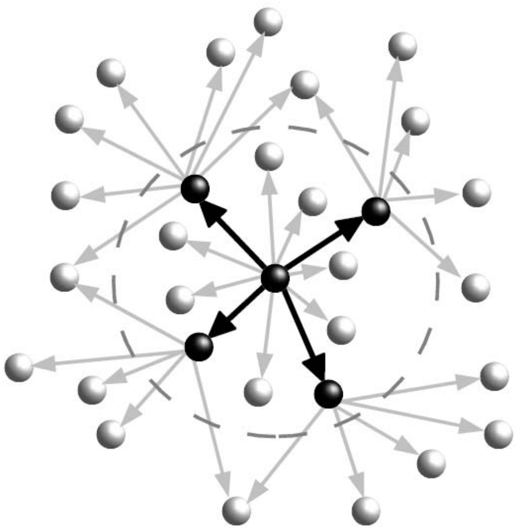

Fig. 1. Multipoint Relays of a node. A node (center) floods a message that is forwarded only by the neighbors it has selected as its MPRs (the black nodes). The range of the neighborhood of the node is depicted by the circle.

3 MADPastry

In this section, we will present a brief overview of MADPastry that provides the DHT functionalities for the DHT-OLSR protocol.
MADPastry [12] is a DHT substrate especially designed for mobile ad hoc networks. It combines AODV ad hoc routing [8] and Pastry overlay routing [10] at the network layer to provide an efficient and low-overhead primitive for key-based routing in MANETs.
For this purpose, each node in a MADPastry network assigns itself a unique virtual ID (for example by hashing its IP address, etc.), which defines its logical position on the virtual overlay ID space [10]. Additionally, each MADPastry message contains a message key (i.e. an ID from the virtual overlay ID space) in its header. MADPastry then routes the message to the node in the network that is currently responsible for the message key – i.e. to the node whose virtual ID is currently the numerically closest to the message key among all MADPastry nodes in the network. To avoid message broadcasts whenever possible (e.g. for route discovery), MADPastry explicitly considers physical locality in the construction of its routing tables.
Clusters. Conventional (Internet-based) DHTs are largely oblivious of the actual physical topology so that two virtual neighbors can be located arbitrarily far from each with respect to the underlying physical network. This can often lead to a large overlay stretch (i.e. the ratio between the length of the physical route traveled during the lookup of a virtual key compared to the direct physical path from the source to the eventual target node) as subsequent overlay hops can literally crisscross the physical network. Due to the volatile nature of physical routes in MANETs, this effect is especially prohibitive in such environments.
To exploit physical locality in the construction and maintenance of its overlay, MADPastry uses Random Landmarking [11]. Instead of having fixed landmark nodes – which simply are not available in MANETs – fixed *landmark keys* are used. These keys divide the virtual overlay ID space into equal sections (e.g. 16 keys with hexadecimal IDs `"0800…000"`, `"1800…000"`, `"2800…000"`, …, `"E800…000"`, `"F800…000"`, etc.). The nodes whose virtual IDs are currently numerically closest to the landmark keys temporarily become landmark nodes and periodically issue beacon messages. Nodes overhear these beacon messages and periodically determine the physically closest temporary landmark node (e.g. in terms of hops). If need be, a node assigns itself a new virtual ID sharing the same prefix with the closest temporary landmark node. It would then (re-)join the network under its new ID. This leads to physically close nodes forming overlay regions, or clusters, with common ID prefixes. In other words, nodes that are close to each other in the virtual overlay ID space are also likely to be close to one another physically. This is demonstrated by Fig. 2 which shows the spatial distribution of virtual ID prefixes in a 250 node MADPastry network. Equal symbols of equal shades represent equal virtual ID prefixes.

Routing Tables. MADPastry maintains three different routing tables: a standard AODV routing table for physical routes from a node to specific target nodes, as well as a sparse Pastry routing

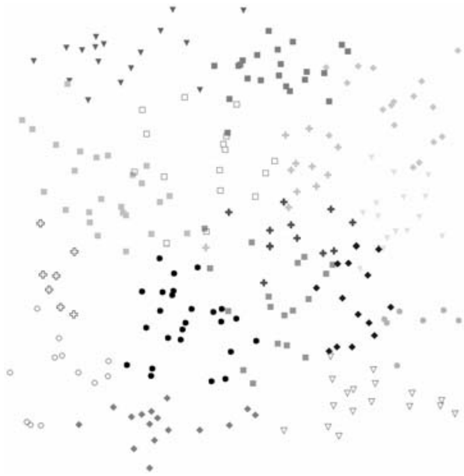

Fig. 2. Spatial distribution of virtual ID prefixes.

table and a standard Pastry leaf set for indirect routing. The Pastry routing table only needs to contains as many entries as are necessary to keep a "finger" entry into each MADPastry cluster (i.e. one entry for each distinct cluster's virtual ID prefix).

Routing Table Maintenance. To avoid the prohibitive overhead induced by routing table maintenance, the only proactive routing table maintenance that a MADPastry node performs is the periodic pinging of its "left" (i.e. the node who has the largest virtual ID smaller than the node's own) and "right" (i.e. the node who has the smallest virtual ID larger than the node's own) leaf as this is necessary to guarantee overlay routing convergence. All other routing entries are gained or updated implicitly by overhearing data packets.

Routing. MADPastry routes packets based on a key. When a node wants to send a packet to a specific key, it consults its Pastry routing and/or leaf set to determine the closest prefix match, as stipulated by standard Pastry. Next, it consults its AODV routing table for the physical route (or, rather, the next physical hop on the route) to execute this overlay hop. Intermediate nodes on the physical path of an overlay hop consult their AODV table for the corresponding next physical hop. When a packet thus reaches the destination of an overlay hop, that node again consults its Pastry routing table and/or leaf set to determine the next overlay hop. This process continues until the packet reaches the eventual target node that is responsible for the packet key – i.e. whose virtual ID is the numerically closest to the packet key.

4 The DHT-OLSR Protocol

In this section, we introduce a novel ad hoc routing protocol called DHT-OLSR. The principle of this protocol is that each node runs OLSR locally within the cluster of nodes that are currently within a certain scope (i.e. a given number of hops). When a node needs to forward a packet towards a destination that is not currently listed in the routing table maintained by OLSR, the node uses a DHT-based Unicast to immediately forward the packet towards those more remote destinations.

For this purpose, each node running DHT-OLSR maintains a regular OLSR routing table. Being a proactive routing protocol – i.e. each node tries to maintain a valid route to every other node currently in the network at all times – OLSR can provide very efficient and low-delay routing

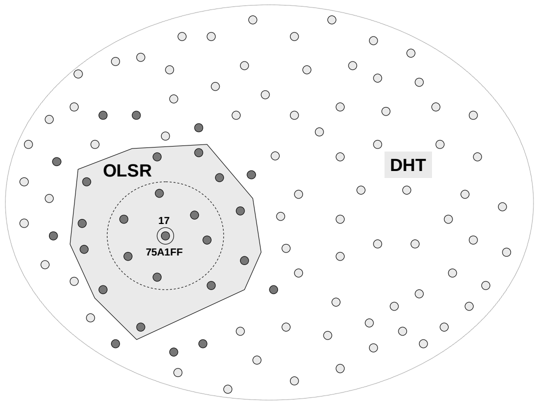

Fig. 3. Visualization of DHT-OLSR's routing concept.

for MANETs. As routes to all potential destinations will usually be known to a node, nodes forward data packets without buffering them. Since nodes in MANETs will likely have quite limited resources, this elimination of the need to queue packets while waiting for route discoveries will often be desirable in MANETs. On the other hand, the apparent downside to this approach is that it will incur an ever larger amount of maintenance traffic as the network size increases. In order to confine this OLSR maintenance traffic in larger networks, DHT-OLSR nodes restrict their OLSR routes to a local scope. This is achieved by simply limiting the TTL of TC messages. For example, nodes might set the TTL of the TC messages that they issue to 2 hops, which effectively places each node at the center of its own OLSR cell with a diameter of 4 hops.
Hence, whenever a node receives (or originates) a data packet, it first tries to lookup the route in its OLSR routing table. If a valid route is found, the data packet is forwarded to the next hop – as stated in the corresponding routing table entry – on the path towards the destination. However, in case no route could be found in the OLSR routing table, the data packet will be forwarded using DHT-style key based routing.
Fig. 3 visualizes the general concept of the DHT-OLSR routing protocol. It depicts node 17 (whose virtual MADPastry ID is `75A1FF`) at the center of its local scope (2 hop radius) OLSR cell. In this example, the grey-shaded area around node 17 shows the 2-hop radius in which node 17's TC messages are propagated (with the dotted circle representing the radio range of node 17). Assuming equal radio ranges, the OLSR routing table of node 17 will contain routes to all nodes within this 2-hop radius (as node 17 trivially lies within the respective OLSR cells of its 1- and 2-hop neighbors), as is depicted by the dark-shaded nodes. Note that node 17's routing table might also contain routes to some (but not necessarily all) of its neighbors three hops away (again depicted in dark grey) as they might be included in the TC messages received from 2-hop neighbors.
As seen, in order to restrict the overhead of OLSR maintenance traffic, DHT-OLSR nodes only run a local scope instance of OLSR. Therefore, whenever a node wants to forward a data packet to a destination for which it cannot find a valid route in its OLSR routing table, the node engages in DHT-based Unicasting. For this purpose, each DHT-OLSR node also runs an instance of MADPastry with its low maintenance overhead to communicate with more remote nodes in the network. However, deviating from the original MADPastry, DHT-OLSR nodes will always drop data packets rather than engage in an AODV route discovery when the route for the current overlay hop is not known. Being a DHT, however, MADPastry provides key-based routing – i.e. packets are no longer routed based on an IP destination address but rather based on a virtual ID (key) from the MADPastry ID space. The question now arises how DHT key-based routing can be used to deliver a packet to its given IP destination address. To resolve IP destination addresses, DHT-OLSR uses a unicast scheme based on the approach presented in [13].

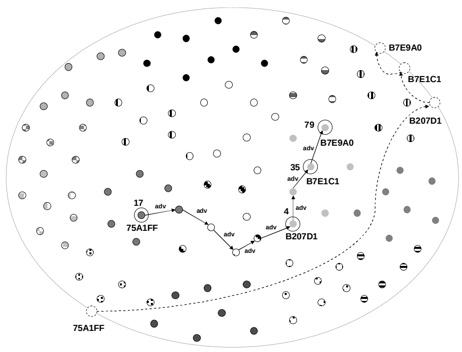

Fig. 4. DHT-OLSR virtual ID advertising. Equal symbols of equal shade represent equal virtual ID prefixes.

In order for nodes to be able to resolve node addresses to their corresponding current virtual IDs, each DHT-OLSR periodically advertises its current virtual ID to the network. For this purpose, a node x hashes its IP address into the virtual overlay ID space to obtain its advertisement key. Using this key, the node will then simply send an advertisement packet containing its current virtual ID towards the node currently responsible for this advertisement key. That node will, upon reception of the advertisement packet, store node x's address along with its current virtual ID.

Node address resolution, now, works in an analogous fashion. Whenever a node y wants to send a data packet to a give destination using the DHT unicast, it needs to resolve the destination node address in order to obtain its current virtual ID. If node y does not know the destination's current virtual ID, it will simply hash the destination node address and acquire the same advertisement key that the destination used to advertise its current virtual ID. Next, node y will send the data packet towards that advertisement key – which by definition (assuming consistent DHT routing) – will be delivered to the node that has previously received the destination node's advertisement. That node will then forward the data packet using the key provided in the advertisement and the data packet will be delivered to the given destination node – as that node will trivially be responsible for its own virtual ID.

Fig. 4 depicts an example of DHT-OLSR's virtual address advertising. Node 17 intends to advertise its current virtual ID `75A1FF`. For this purpose it simply hashes its IP address, which yields the following advertisement key: hash(17) → `B7E97D`. Using MADPastry, node 17 will then send an advertisement packet containing its current virtual ID towards the node currently responsible for its advertisement key. In the first overlay hop, MADPastry routing delivers the packet to node 4 with virtual ID `B207D1`. In the next overlay hop, node 4 forwards the packet on to node 35 (virtual ID `B7E1C1`) who, then, forwards the packet to node 79 whose virtual ID `B7E9A0` is numerically closest to node 17's advertisement key (`B7E97D`) among all other nodes' virtual IDs. Thus, node 79 is responsible for the packet and stores the tuple {17, 75A1FF} in its cache.

Fig. 5 visualizes how DHT-OLSR nodes use DHT unicasts to forward data packet to remote nodes. In this example, node 51 wants to send a data packet to node 17. After discovering that there is no valid route to destination 17 in its OLSR routing table, node 51 engages in a DHT unicast to deliver the data packet. Since node 51 does not know node 17's current virtual ID, it hashes the destination address into the MADPastry virtual ID space and, by definition, acquires the same advertisement key as node 17 did for its advertisement in the example above: hash(17) → `B7E97D`. Next, node 51 forwards the data packet towards that key. Assuming consistent DHT routing, this will deliver the packet to the same node that has previously received node 17's advertisement: in the first overlay hop, the packet is routed to node 42 with virtual ID `B7E705`, who then forwards the packet on to node 79 (virtual ID `B7E9A0`). Obviously, node

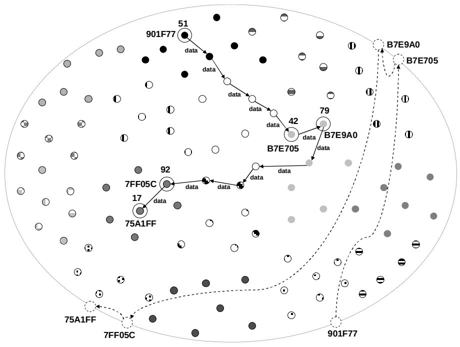

Fig. 5. DHT OLSR virtual ID resolution and data forwarding. Equal symbols of equal shade represent equal virtual ID prefixes.

79 can now resolve node 17's address and then sends the packet towards node 17's current virtual ID. As node 17 is trivially responsible for its own virtual ID, the data packet will, thus, be delivered over node 92 (virtual ID `7FF05C`) to the original destination (node 17).
By using the hybrid mechanisms described in this section, a particular feature of DHT-OLSR is that it drastically reduces the impact of remote control traffic on the bandwidth available locally. Since MADPastry organizes the nodes into virtual clusters of physically close nodes, control traffic generated by MADPastry will generally not affect nodes in other clusters. Therefore, a node will mostly only have to bear the OLSR control traffic limited to the given hop radius and the leaf set control traffic from its local MADPastry cluster.

5. Evaluation of the DHT-OLSR Protocol

In order to evaluate the general feasibility of our approach, we have implemented DHT-OLSR in the network simulator ns-2. In order to put the performance of DHT-OLSR into perspective, its results were compared to that of a conventional proactive – OLSR – and reactive routing protocol – AODV (we are using the AODV-UU implementation 0.9.1 for ns-2.). For all simulations conducted, all nodes were moving around according to the Random Waypoint model at a constant velocity of 1.4 m/s (quick walking pace) with 0s pause time (constant node mobility). Nodes were communicating with each other in an ad hoc mode using the 802.11 communication standard with a transmission range of 250m. We evaluated different network sizes of 50, 100, 150, 200, and 250 nodes while keeping the node density fixed at 100 nodes per square kilometer.
We have analyzed the following two metrics:
Packet Delivery Ratio (PDR). This figure represents the percentage of data packets that are successfully delivered to their respective destinations.
Total Amount of Traffic. Since many different packet types (e.g. AODV route requests, MADPastry packets, OLSR control packets, etc.) of various packet lengths are transmitted during a simulation run, we are not evaluating the total packet count. Instead, we are considering the total network traffic (in kilo bytes) that is created during the simulated hour. Whenever a node forwards a packet, this figure is increased by the packet size. This figure includes *all* routing and application level packet types (data packets, control packets, etc.).
To evaluate the performance of the different routing protocols, we chose a very simple traffic pattern. In our simulations, each node sends a data packet to a randomly picked destination every 10 seconds. Each run simulated the course of one hour.
Fig. 6 shows the PDR achieved by the respective routing protocols for various network sizes. As can be seen, DHT-OLSR's PDR decreases only slightly with an increasing network size and

remains at or above 90% in all scenarios considered. Both the conventional reactive and the proactive routing protocol, however, do not scale well to large network sizes and see their re-

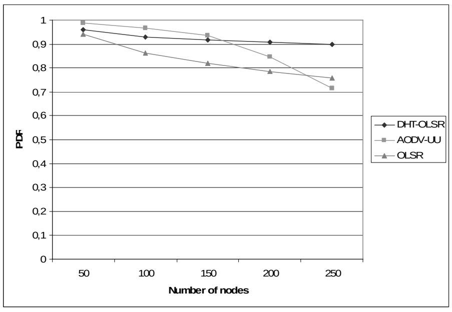

Fig. 6. Packet delivery ratios vs. number of nodes.

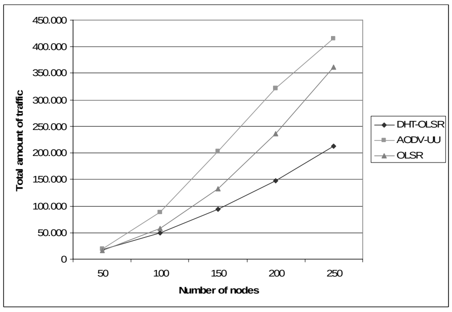

Fig. 7. Total traffic vs. number of nodes.

spective PDRs fall drastically as the number of nodes in the network increases. The reason for this that DHT-OLSR mostly uses local routes (also for MADPastry overlay routing, see [13]) that are updated through local control traffic (as explained in the previous section) whereas both AODV's and OLSR's control traffic affects the entire network, which consumes more and more bandwidth as the network size increases.
This effect also becomes clear in Fig. 7 which shows that DHT-OLSR also produces significantly less overall traffic than the two reference protocols do. Again, this is due to the fact that DHT-OLSR nodes generally are only affected by local control traffic whereas both OLSR's and AODV's control packets are broadcast.

6. Related Work

There have been a number of hybrid ad hoc routing protocols proposed such as ZRP [6]. These protocols are based on various combinations of proactive (for local scope routing) and reactive routing (for more remote destinations). A whole plethora of protocols have been proposed with OLSR [4] being a typical proactive protocol and AODV [8] a very popular reactive one. However, it is clearly beyond the scope of this paper to exhaustively review all of those. In this paper, we will restrict ourselves to the protocols that are considered for standardization in the IETF: one proactive approach and one reactive approach that are not designed to scale to large MANET topologies. Hence, neither hybrid combination of those approaches considered by the IETF can be expected to scale.
For key-based routing in MANETs, a few approaches have been proposed recently, including Ekta [7], SSR [5], VRR [2], and MADPastry [12]. We have chosen the latter for its convenient property of explicitly considering physical locality in the construction of its routing tables.

7. Conclusion

In this paper, we have introduced a scalable ad hoc IP routing protocol, fit for large and mobile MANET topologies, based on a hybrid scheme mixing ideas coming from both the mobile ad hoc routing field, and the peer-to-peer networking field. Simulations so far show that DHT-OLSR outperforms the protocols standardized by the IETF in a variety of large MANET topologies.
Future work on DHT-OLSR will include further evaluations of the protocol, both through simulations and real physical testbed implementations. Here, the performance of DHT-OLSR should

also be compared against more ad hoc routing protocols such as the hybrid ZRP [6] or Fisheye OLSR [1]. Potential enhancements to DHT-OLSR are envisioned such as mechanisms to dynamically adjust OLSR parameters or to adapt the number of MADPastry clusters.
DHT-OLSR uses efficient key-based unicast routing to scale to very large MANET topologies along with OLSR routing to naturally integrate with the Internet infrastructure. Therefore, this protocol provides an architecture that may introduce a gradual transition from traditional IP routing towards scalable IP MANET routing.

References

1. M.C. Adjih, E. Baccelli, T.Clausen, P. Jacquet, R. Rodolakis. "Fish Eye OLSR Scaling Properties". *IEEE Journal of Communication and Networks, Special Issue on Mobile Ad Hoc Wireless Networks*, December 2004.
2. M. Caesar, M. Castro, E. Nightingale, G. O'Shea, and A. Rowstron. "Virtual Ring Routing: Network routing inspired by DHTs". In *Proc. of ACM SIGCOMM'06*, September 2006.
3. I. Chakeres and C. Perkins. "Dynamic MANET On-demand (DYMO) Routing", IETF Internet draft (work in progress), http://www.ietf.org/internet-drafts/draft-ietf-manet-dymo-06.txt, October 2006.
4. T. Clausen, P. Jacquet Ed. "RFC 3626: Optimized Link State Routing Protocol," Internet Engineering Task Force (IETF) RFC, http://ietf.org/rfc/rfc3626.txt, 2003.
5. T. Fuhrmann. "Scalable Routing for Networked Sensors and Actuators". In Proc. of SECON 2005, September 2005.
6. Z. J. Haas, M. R. Pearlman, and P. Samar. "The Zone Routing Protocol (ZRP) for Ad Hoc Networks". Internet-Draft, http://tools.ietf.org/wg/manet/draft-ietf-manet-zone-zrp/draft-ietf-manetzone-zrp-04.txt, July 2002.
7. H. Pucha, S. M. Das, and Y. C. Hu. "Ekta: An Efficient DHT Substrate for Distributed Applications in Mobile Ad Hoc Networks". In *Proc. of WMCSA 2004*, December 2004.
8. C. Perkins, E. Belding-Royer, S. Das. "Ad hoc On-Demand Distance Vector (AODV) Routing", RFC 3561, http://www.ietf.org/rfc/rfc3561.txt, July 2003.
9. A. Qayyum, L. Viennot, A. Laouiti, "Multipoint Relaying: An Efficient Technique for Flooding in Mobile Wireless Networks," INRIA Research Report RR-3898, March 2000.
10. A. Rowstron and P. Druschel. "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems". In *Proc. of Middleware*, November 2001.
11. R. Winter, T. Zahn, and J. Schiller. "Random Landmarking in Mobile, Topology-Aware Peer-to-Peer Networks". In *Proc. of FTDCS*, May 2004.
12. T. Zahn and J. Schiller. "MADPastry: A DHT Substrate for Practicably Sized MANETs". In *Proc. of ASWN*, June 2005.
13. T. Zahn, and J. Schiller. "DHT-based Unicast for Mobile Ad Hoc Networks". In *Proc. of MP2P'06*, March 2006.